

Reservoir Operation Using a Robust Evolutionary Optimization Algorithm

Jafar Y. Al-Jawad^a, Tiku T. Tanyimboh^{b*}

^{a,b}Department of Civil and Environmental Engineering, University of Strathclyde Glasgow 75
Montrose St, Glasgow G1 1XJ

^ajafar.al-jawad@strath.ac.uk; ^btikutanyimboh@hotmail.co.uk

*Corresponding author

Citation: Al-Jawad JY, Tanyimboh TT (2017) Reservoir operation using a robust evolutionary optimization algorithm. Journal of Environmental Management, Volume 197, Pages 275–286. DOI: <https://doi.org/10.1016/j.jenvman.2017.03.081>

The final publication is available at:

<http://www.sciencedirect.com/science/article/pii/S0301479717302979>

Reservoir Operation Using a Robust Evolutionary Optimization Algorithm

Abstract

In this research, a significant improvement in reservoir operation was achieved using a state-of-the-art evolutionary algorithm named Borg MOEA. A real-world multipurpose dam was used to test the algorithm's performance, and the target of the reservoir operation policy was to fulfil downstream water demands in drought condition while maintaining a sustainable quantity of water in the reservoir for the next year. The reservoir's performance was improved by increasing the maximum reservoir storage by 14.83 million m³. Furthermore, sustainable water storage in the reservoir was achieved for the next year, for the simulated low flow condition considered, while the total annual imbalance between the monthly reservoir releases and water demands was reduced by 64.7%. The algorithm converged quickly and reliably, and consistently good results were obtained. The methodology and results will be useful to decision makers and water managers for setting the policy to manage the reservoir efficiently and sustainably.

Keywords: Evolutionary optimization algorithm; reservoir operation policy; multipurpose reservoir system; reservoir drawdown limits; self-adaptive recombination; environmental water management

1. INTRODUCTION

Multipurpose reservoirs are widely used to serve multiple demands for domestic, industrial, irrigation, environment, hydropower production and flood control, to maximize the economic benefits. These types of systems are complex because of the nonlinear storage-inflow relationship, conflicting objectives, dynamic properties, nonlinear constraints, etc. (Haimes and Hall 1977). In the field of water resources management, significant demands on water exploitation were observed in recent decades. This raises the challenge to manage and allocate water in a sustainable way, and reservoirs are essential for water resources management in a river basin (Horne *et al.* 2016, Jothiprakash and Shanthi 2006) .

Many methods for optimization were found to solve different types of problems such as linear programming, non-linear programming and dynamic programming, etc. (Horne *et al.* 2016). However, the classical optimization methods are generally not suitable for such complex problems for a number of reasons. For example, typically, they provide a single local optimum solution. Evolutionary algorithms on the other hand, use a population of solutions rather than one solution in every iteration (Deb 2001). In recent decades, evolutionary optimization algorithms were widely used in different fields of engineering and science to solve real-world problems (Coello *et al.* 2007).

Regarding engineering applications, Formiga *et al.* (2003) used the Non-dominated Sorting Genetic Algorithm (NSGA II) to solve water distribution network problems. Régnier *et al.* (2005) applied NSGA II in electromechanical system design. In structural design, Tract (1997) used a genetic algorithm (GA) with Pareto ranking in truss design. Deb and Tiwari (2005) used NSGA II for design in the field of mechanical engineering. In the field of civil engineering, Feng *et al.* (1999) used a GA with Pareto ranking to optimize building construction planning.

To achieve effective operational management policies for water resources

management problems, many researchers used different optimization approaches (Horne *et al.* 2016). Sharif and Wardlaw (2000) used a GA to maximize the hydropower production while allowing deficits to occur in irrigation supplies. Chenari *et al.* (2014) also used a GA to determine the releases from a reservoir. Furthermore, Tilmant *et al.* (2002) used fuzzy stochastic dynamic program to optimize the control rules for a multipurpose reservoir. Kim and Heo (2006) used MOGA (multi-objective genetic algorithm) to solve a multi-reservoir multi-objective problem. Wu and Zou (2012) applied MOGA to maximize both power generation and irrigation benefits. Scola *et al.* (2014) applied NSGA II to maximize power generation. Cancelliere *et al.* (2003) used a multi-objective optimization method to reduce the deficit in the releases for irrigation and improve municipal volumetric reliability.

Borg MOEA is a recent optimization algorithm that was introduced by Hadka and Reed (2013). In this research, Borg MOEA was used to solve a reservoir operation problem. These types of problems need a powerful algorithm to handle the complexity of the inflow-storage relationship. The Borg MOEA algorithm has six operators that compete to create offspring in each generation. The effectiveness of the algorithm is maintained throughout the optimization by deploying the most suitable combination of operators for crossover. In addition, Borg MOEA is able to detect stagnation and escape from local optima by reviving the search process.

The aim of the current study was to investigate the robustness and performance of the algorithm on a reservoir operation problem. A drought condition and an additional reservoir drawdown constraint were considered in order to test the algorithm's ability to find good solutions consistently in such critical conditions. In reservoir management, it is difficult to control the releases over the entire year in order to fulfil the downstream demands and to maintain the same or higher initial water storage in the reservoir for the next year in drought conditions. Hence, the influence of the extra drawdown constraint imposed was investigated.

2. OVERVIEW OF THE OPTIMIZATION APPROACH

Hadka and Reed (2013) introduced Borg MOEA for many-objective optimization problems. Some of the features in Borg MOEA include (a) diversity preservation; (b) measurement of search progress and stagnation; (c) restart to move away from local optima; (d) multiple recombination operators that compete to produce offspring; and (e) use of a dominance archive. The algorithm uses six operators in the recombination process to improve the search progress and a dominance archive to store all the non-dominated solutions.

To preserve diversity, the objective space is divided into hyper-boxes whose dimensions are all equal to ϵ , as in Figure 1. Thus the ϵ -box index vector is used to find the dominant solutions instead of the objective function values. The algorithm calculates this index by dividing the objective function value by ϵ , and then sets the result as the succeeding integer value. If two or more solutions are in the same ϵ -box, the dominant solution is the one which is nearest to the lower-left corner of the ϵ -box, in the case of a minimization problem.

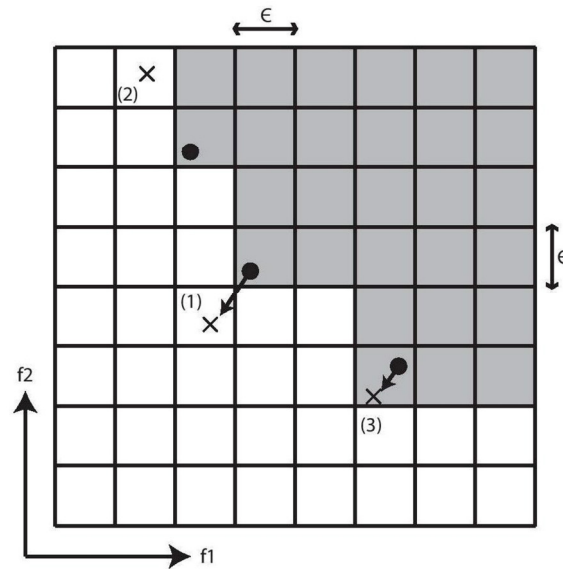


Figure 1. Graphical representation of the ϵ -progress concept in a minimization problem with two objectives. Solutions (1) and (2) are new solutions in unoccupied boxes and thus represent improvements. Solution (3) is not considered as an improvement because it resides in a previously occupied box. The shaded boxes were previously occupied while the unshaded boxes were not previously occupied (Hadka and Reed 2013).

For stagnation measurement, ϵ -progress was introduced, which measures the improvement while searching for new solutions. If the algorithm finds new solutions in a new unoccupied ϵ -box, it means that there is progress and the algorithm is allowed to continue. This can be observed more clearly in Figure 1. On the other hand, if there is no improvement based on ϵ -progress for a certain number of evaluations, a revival process is triggered, to escape from any local optima. The details of the restart procedure are available in Hadka and Reed (2013). The algorithm maintains the population size as a certain ratio of the archive size during the optimization process. This feature was adopted from ϵ -NSGA II (Kollat and Reed 2006) and is called the injection rate.

The algorithm employs multiple recombination operators to produce offspring. In fact, Borg MOEA provides a framework in which the selection of the recombination operators adjusts depending on the dynamic properties of the objective and solution spaces of the optimization problem, including the make-up and diversity of the candidate solutions, and the landscape of the objectives. The recombination operators in Borg MOEA are:

- (a) simulated binary crossover (SBX) (Deb and Agrawal 1994);
- (b) differential evolution (DE) (Storn and Price 1997);
- (c) parent-centric crossover (PCX) (Deb *et al.* 2002);
- (d) unimodal normal distribution crossover (UNDX) Kita *et al.* (2000);
- (e) simplex crossover (SPX) (Tsutsui *et al.* 1999); and
- (f) uniform mutation (UM) (Michalewicz *et al.* 1994).

Also, the polynomial mutation (PM) (Deb and Agrawal 1994) is applied to the offspring produced by all the operators except for UM.

The probability of choosing a particular recombination operator to produce offspring depends on its ability to contribute nondominated solutions in the dominance archive, compared to the other operators; hence the operator selection probabilities are proportional to

their effectiveness and respective contributions.

The values of the decision variables in the offspring generated lie within the upper and lower bounds of the decision variables. The algorithm has many coefficients and parameters as summarised in Table 1 (Hadka and Reed 2013) in which L represents the number of decision variables, ϵ is the dimension of the hyper-box in the objective space, and σ_η , σ_ξ and σ_ζ represent the variance parameters that control the spatial distribution of the resulting offspring for the PCX and UNDX operators.

The values of the parameters shown in Table 1 are the recommended empirical values from the literature, based on extensive testing that included complex real-world problems (Hadka *et al.* 2012, Reed *et al.* 2013). The values have been used widely in subsequent studies also (Zheng *et al.* 2016). Further improvement through fine tuning on a case-specific basis may be possible; however this issue is complex (Lobo *et al.* 2007) and is not the main focus of the present research.

Hadka and Reed (2012) presented comparisons of eight state-of-the-art evolutionary algorithms based on their performance on eight test functions. Furthermore, Reed *et al.* (2013) compared the performance of Borg MOEA on real-world water resources problems with ten competitive algorithms. Both studies concluded that Borg MOEA outperformed the other algorithms on the problems considered.

The criteria used in the comparisons included the hypervolume, generational distance and additive ϵ -indicator metrics (Knowles and Corne 2002). For a set of nondominated solutions, the hypervolume represents the fraction of the objective space that the solutions dominate. It increases as: the solutions approach the Pareto-optimal front; their range increases; and their distribution becomes more even. The generational distance calculates the average distance between the resulting nondominated front and the Pareto-front. The additive ϵ -indicator measures the smallest factor by which the resulting approximation set achieved

must be translated in the objective space in order weakly to dominate the reference set. Based on these criteria, the main conclusion was that Borg MOEA showed significant advantages over the other algorithms.

Table 1 Default values of the parameters used in Borg MOEA

Parameter	Value	Parameter	Value
Initial population size	100	SPX parents	10
Tournament selection size	2	SPX offspring	2
Epsilon, ϵ	0.01	SPX epsilon	2.0
SBX rate	1.0	UNDX parents	10
SBX distribution index	15.0	UNDX offspring	2
DE crossover rate	1.0	UNDX σ_ξ	0.5
DE step size	3.0	UNDX σ_η	$0.35/\sqrt{L}$
PCX parents	10	UM rate	$1/L$
PCX offspring	2	PM rate	$1/L$
PCX σ_η	0.1	PM distribution index	20
PCX σ_ζ	0.1		

ϵ is the dimension of hyper-boxes in objective space; L is the number of decision variables; and the various σ symbols are variance-related parameters.

The algorithms considered by Hadka and Reed (2013) in their comparative study are listed below, with additional details in Deb *et al.* (2003), Zhang *et al.* (2009), Sierra and Coello Coello (2005), Kollat and Reed (2006), etc.

- a. ϵ -MOEA
- b. MOEA/D (multi-objective evolutionary algorithm based on decomposition)
- c. GBE3 (generalized differential evolution, version 3)
- d. OMOPSO (multi-objective particle swarm optimization)
- e. IBEA (indicator-based evolutionary algorithm)
- f. ϵ -NSGA II

In another study, in addition to the previous algorithms, Reed *et al.* (2013) compared Borg MOEA based on four test problems with NSGA II (Deb *et al.* 2002), SPEA2 (Zitzler *et al.* 2002) and AMALGAM (Vrugt and Robinson 2007). The authors concluded that Borg MOEA was the best among the nine algorithms, including a. to f. in the preceding list.

3. RESERVOIR OPTIMIZATION MODEL

Usually, multipurpose reservoirs serve many goals like hydropower generation, domestic water supply, agricultural water supply, flood protection, and other environmental goals. In this study, the reservoir system consists of a single multipurpose dam constructed to control water discharge in the river for irrigation and domestic use, flood control and hydropower generation. This type of dam has many economic benefits. In this model, three types of constraints were considered. A drought condition was considered in order to test the algorithm's ability to find an optimum solution in such critical conditions without violating the reservoir drawdown limit imposed.

3.1 Reservoir Storage Constraints

The volume of storage in the reservoir is limited between the dead storage and the maximum capacity of the reservoir. The dead storage constraint, which is the minimum allowable storage in the reservoir, is

$$C_1(t) = S_t - S_{min} \geq 0; \forall t \quad (1)$$

where $S_t \geq 0$ is the initial storage at the beginning of the month t , $t = 1, \dots, 12$; S_{min} is the dead storage of the reservoir. The maximum storage constraint, which is the maximum storage capacity of the reservoir, is

$$C_2(t) = S_{max} - S_t \geq 0; \forall t \quad (2)$$

where S_{max} is the maximum normal storage in the reservoir.

3.2 Reservoir Release Constraints

The releases from the reservoir should be bounded between the minimum and maximum releases. The minimum release constraint, for the minimum amount of water to be released from the reservoir, is

$$C_3(t) = R_t - R_{min} \geq 0; \forall t \quad (3)$$

where $R_t \geq 0$ is the mean monthly water release for month t . R_{min} is the minimum allowable water releases from the reservoir. The maximum allowable amount of water released from the reservoir should not exceed e.g. the spillway or downstream channel capacity. Thus

$$C_4(t) = R_{max} - R_t \geq 0; \forall t \quad (4)$$

where R_{max} is the maximum allowable release from the reservoir.

3.3 Constraint on Annual Reservoir Drawdown

To ensure reservoir storage sustainability, an extra constraint was introduced in this study so that the amount of storage in the first month of the next year will equal or exceed the initial storage of the first month. This constraint can be expressed as

$$C_5(13) = S_{13} - S_1 \geq 0 \quad (5)$$

where S_1 is the initial storage in the first month and S_{13} is the reservoir storage at the start of the first month of the next year.

3.4 Low Reservoir Inflow Condition

A drought condition was considered in order to test the algorithm's ability to find good solutions quickly and consistently in such critical conditions. To calculate this condition, 50% of the standard deviation of the monthly average inflow for many years was subtracted from the original inflow (Reddy and Kumar 2006).

$$I_t = I'_t - 0.5\sigma_t \quad (6)$$

where I_t is the reduced reservoir inflow for month t ; I'_t is the original reservoir inflow for month t ; and σ_t is the standard deviation of the reservoir inflow for month t .

2.5. Fitness Function

The monthly flow continuity equation is

$$S_t + I_t - S_{t+1} - R_t - E_t = 0 \quad (7)$$

where S_{t+1} is the final storage at the end of month t and E_t is the mean monthly evaporation from the reservoir during the month t .

The fitness function for reservoir operation that should be minimized can be expressed as

$$f = \left[\sum_{t=1}^{12} (R_t - D_t)^2 + \sum_{t=1}^{12} (S_t - S_{t+1} + I_t - R_t - E_t)^2 \right] (1 + C)^e \quad (8)$$

where D_t is the mean monthly downstream water demand for the month t ; C is a penalty for constraint violations; and the value of the exponent, e , is 2. The first part of Equation 8 aims to minimize the differences between the monthly reservoir releases and the demands, subject to the flow continuity equation in Eq. 7. The second part is a quadratic penalty function to address constraint violations. At the solution, the continuity equation in Eq. 7 is equal to zero. Also, the constraint violation penalty C is zero for feasible solutions.

Thus the fitness function, Eq. 8, aims to minimize the total annual imbalance between the monthly reservoir releases and water demands, including deficits and surpluses. Self-evidently a deficit implies a shortfall in the supply, while a surplus is to be avoided if possible, as a water conservation measure during periods with low reservoir inflows.

In general, the convergence rate and optimality of the solutions achieved are influenced by the penalty function employed, and the effects differ from a problem to another. Therefore, this function should be chosen carefully for each problem (Deb and Datta 2013, Dridi *et al.* 2008, Saleh and Tanyimboh 2013, 2014, 2016, Siew and Tanyimboh 2012, Siew *et al.* 2014). The problem of formulating and calibrating penalty functions is complex (Chang *et al.* 2010, Coello Coello 2002, Deb and Datta 2013). A review of constraint handling in evolutionary algorithms is available in Coello Coello (2002).

The constraint violation penalty adopted here is

$$C = A(g_1 + g_2 + g_3) \quad (9)$$

where A is a coefficient that was taken as 100, and g_1 , g_2 , and g_3 represent the penalties for the minimum, maximum, and sustainable storage constraints, respectively.

The values of the penalty factor A and exponent e were determined empirically, to apply an appropriate amount of selection pressure that would not render all the infeasible solutions including those with relatively small constraint violations totally uncompetitive (Dridi *et al.* 2008, Tanyimboh and Seyoum 2016, Yang and Soh 1997). Indeed, evolutionary algorithms that include nondominated or competitive infeasible solutions in the optimization process generally achieve better results than those that fail to exploit any infeasible solutions generated (Barlow and Tanyimboh 2014, Eskandar *et al.* 2012, Yang and Soh 1997, Woldesenbet *et al.* 2009, Siew *et al.* 2016).

On the other hand, an algorithm's convergence rate may be too slow if the selection pressure is insufficient. For example, Siew and Tanyimboh (2012) compared two versions of a performance function that represents the fitness. They adopted the version with more selection pressure and significantly faster convergence.

The penalties for violating the minimum, maximum, and sustainable storage constraints, respectively, are as follows.

$$g_1 = \sum_{t=1}^{12} \text{Max}[0, (S_{\min} - S_t)] \quad (10)$$

$$g_2 = \sum_{t=1}^{12} \text{Max}[0, (S_t - S_{\max})] \quad (11)$$

$$g_3 = \text{Max}[0, (S_1 - S_{13})] \quad (12)$$

The form of fitness function adopted in Eq. 8 has the advantages that it allows simultaneous minimization of both the objective and penalty functions. The penalty function is dynamic and reflects the degree of constraint violation. This allows promising infeasible solutions to contribute essential genetic material to the gene pool. The quadratic form of the penalty function adjusts the selection pressure on the infeasible solutions gradually as the optimization progresses (Yang and Soh 1997), thus shifting the emphasis of the search

progressively away from more exploration at the start to more exploitation at the end.

The formulation of the penalty function aims to exploit all the solutions generated fully, including virtually feasible solutions that promote exploration and exploitation around the active constraint boundaries. In this way, the whole solution space is searched effectively. The infeasible solutions enhance diversity, promote active boundary search, help avoid a purely interior search and premature convergence, and improve the overall effectiveness of the algorithm (Siew and Tanyimboh 2012, Yang and Soh 1997).

The total number of decision variables is 25, i.e. 12 for the monthly releases and 13 for the storages as shown in Figure 2b and 2c, in which the 13 month represents the first month of the following year. We wrote a computer program in C++ language to solve the optimization problem in Equations 1 through 12 using Borg MOEA. The algorithm was executed ten times with 200,000 function evaluations allowed in each run, with an initial population of 100, for each scenario of the optimization problem.

The period of operation considered was one year. For long-term planning, the number of decision variables and dimensionality of the problem may increase and/or longer time steps may be used. Monthly rather weekly values were considered in the model as the focus of the research is to assist with the development of an efficient seasonal operating policy, rather than daily operational control (Horne *et al.* 2016). Environmental water management decisions may relate to a range of spatial and temporal scales, from sub-daily to multi-year and a single location to the river basin, respectively. Horne *et al.* (2016) mentioned the importance of the relationships between the various scales and provided examples of the strategies used such as nested and hierarchical models, and stochastic programming.

In addition, seepage from the reservoir and other operational losses were neglected, based on the problem specification in Chenari *et al.* (2014). These issues are not the main focus of the present investigation; a simulation model that provides the relevant properties of

the system (i.e. inflow, evaporation, etc.) could be used instead if necessary. Hence the losses may be incorporated, with additional case-specific data.

4. ILLUSTRATIVE EXAMPLE

A real-world case study from the literature was adapted in this study. Chenari *et al.* (2014) employed a GA to optimize the reservoir operation for Mahabad dam in Iran. The aim was to minimize the deficit in water demands. The dam, located in the northwest of Iran, has an approximate watershed area of 807 km². It is in a cold semi-arid area with average annual rainfall of 542.58 mm. There is rainfall during the three months from February to April. The live storage and dead storage are 180 million m³ and 40 million m³, respectively. The minimum release was taken as zero and the maximum release was taken as 51.48 million m³ per month for the first six months of the year and 53.57 million m³ per month for the second six months of the year. Data for 32 years, from 1975 to 2006, were used by Chenari *et al.* (2014) to obtain the average monthly inflows to the reservoir. Table 2 presents the values of inflows and water demands in the case study area.

Table 2 Reservoir inflows and water demands (Chenari *et al.* 2014)

Month	Average inflow (10 ⁶ m ³)	Standard deviation (10 ⁶ m ³)	Drought season inflow (10 ⁶ m ³)	Water demand (10 ⁶ m ³)	Maximum release (10 ⁶ m ³)
September	1.340	1.450	0.615	20.67	51.84
October	7.850	11.86	1.920	9.110	51.84
November	11.03	11.33	5.365	1.530	51.84
December	16.28	15.30	8.630	1.430	51.84
January	20.98	14.36	13.80	1.400	51.84
February	54.00	33.26	37.37	1.440	51.84
March	97.13	43.28	75.49	6.290	53.57
April	55.88	37.70	37.03	27.04	53.57
May	10.90	10.80	5.500	33.01	53.57
June	2.470	1.870	1.535	29.64	53.57
July	1.140	0.940	0.670	30.74	53.57
August	0.900	0.920	0.440	26.80	53.57

More details and data can be found in Chenari *et al.* (2014) that used a population size of 350 and 1500 generations, i.e. 525,000 function evaluations. The final value of the objective function in Chenari *et al.* (2014) was $185.3 \times 10^6 \text{ m}^3$. The minimum and maximum storage in the reservoir were 49.99 million and 165.17 million m^3 , respectively.

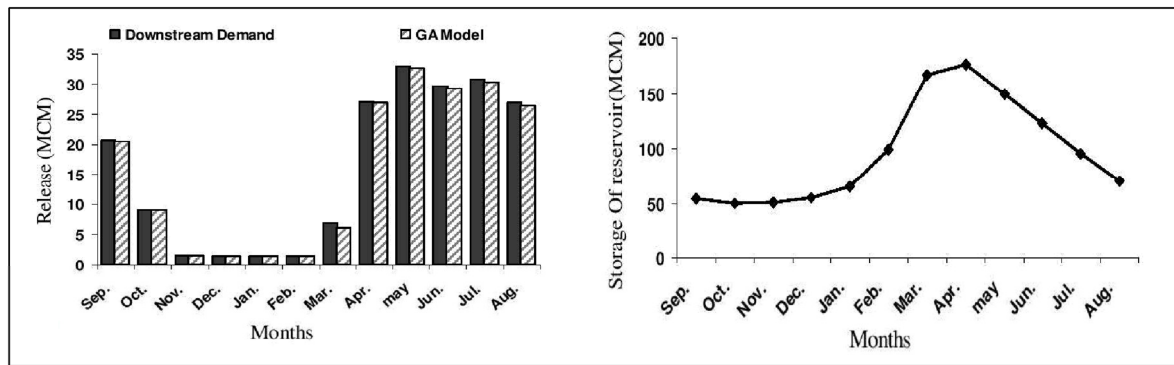
5. RESULTS AND DISCUSSION

4.1 Reservoir Storage and Release

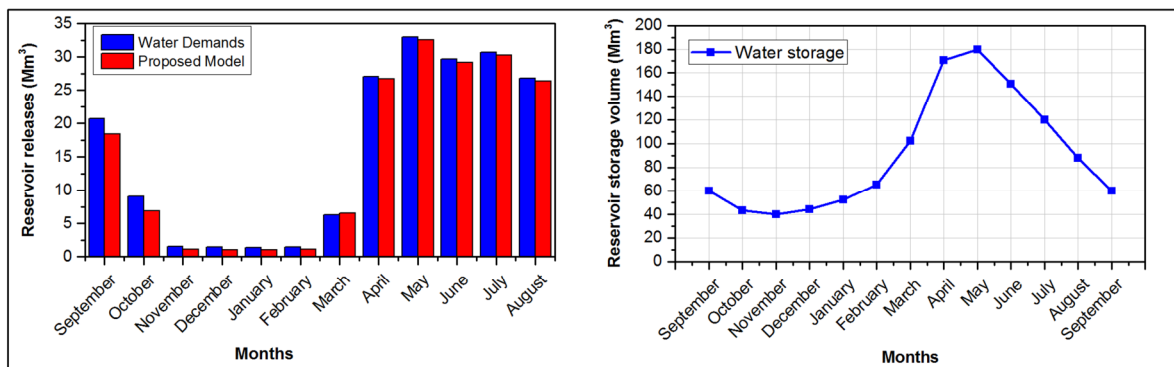
Figure 2a illustrates the monthly reservoir releases and storage reported by Chenari *et al.* (2014) while Figure 2b and 2c show the corresponding results achieved in this study. The initial storage in the reservoir was insufficient; hence some deficits occurred especially in the first two months (September and October). Also, the effect of the water sustainability constraint on the releases is observed especially in the last five months (April to August), which causes some deficits in the releases due to this constraint (Figure 2b).

The sustainability constraint was not considered in the original formulation of the problem in Chenari *et al.* (2014), and Figure 2c shows the results achieved in this study for the original problem specifications in Chenari *et al.* (2014). In Figure 2a, the first six months (September to February) show a good match between the releases and demands. However, these results do not match the reservoir storage shown in the storage graph.

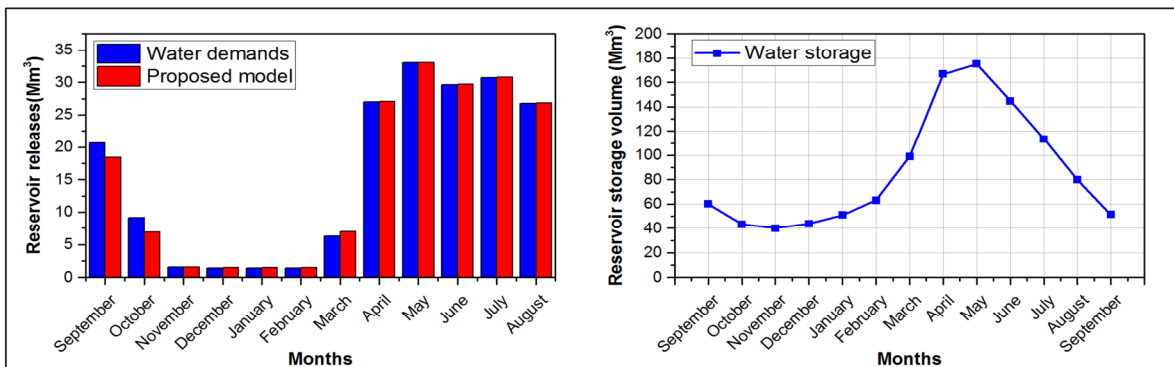
Moreover, Figure 2a shows that the initial storage in September is about 60 million m^3 , and the releases from the reservoir in the same month is about 20 million m^3 , and inflow is $0.615 \times 10^6 \text{ m}^3$ (Table 2). It means that in the next month the water storage in the reservoir will be approximately equal to the dead storage, i.e. 40 million m^3 . Then, the release in the next month is 9.11 million m^3 and the inflow is 1.92 million m^3 (Table 2). This means that the storage in the reservoir will be less than the dead storage. Consequently, there should be deficits in the releases for the first two months to maintain the water storage limits in the reservoir as observed in Figure 2b and 2c.



(a)



(b)



(c)

Figure 2. Reservoir releases and storage (a) Chenari *et al.* (2014) (b) Present formulation with sustainability constraint (c) Present formulation without sustainability constraint

5.2. Decision Variable Values Achieved

Table 3 summarises the reservoir release, deficit and storage for each month based on ten runs of the optimization algorithm. The maximum standard deviation of the releases was 0.151 million m^3 in November and the minimum was 0.004 million m^3 in September. There were deficits in all months except for March. The deficits arose because the amount of water in the reservoir and the inflows could not fulfil the water demands and evaporation losses.

The deficit over the entire year occurs because of the sustainability constraint adopted in this research. This constraint ensures the storage in the beginning of the next year will be equal or larger than the initial storage in the current year. For planning and operational purposes, more sophistication of the reservoir depletion constraint may be required to optimise the benefits further as, in practice, short-term drawdown of the reservoir would likely be acceptable. For example, Kim and Heo (2006) used smaller ranges of upper and lower storage limits than the original limits for the next year. Chang *et al.* (2010) allowed depletion between the initial and next year's storage of 10%. The effect of the sustainability constraint is examined further in Subsection 5.5 based on the original problem specifications in Chenari *et al.* (2014).

Table 3 Reservoir operation results based on ten optimization runs

Month	Release ($\text{m}^3 \times 10^6$)			Deficit ($\text{m}^3 \times 10^6$)		Storage ($\text{m}^3 \times 10^6$)		
	Minimum	Mean	Std.	Minimum	Mean	Minimum	Mean	Std.
September	18.431	18.439	0.004	2.224	2.231	60.000	60.000	0.000
October	6.839	6.875	0.013	2.215	2.235	43.184	43.202	0.011
November	1.132	1.270	0.151	0.007	0.260	40.000	40.000	0.000
December	1.040	1.164	0.131	0.076	0.266	43.874	44.363	0.288
January	1.031	1.115	0.095	0.117	0.285	51.258	52.102	0.538
February	1.092	1.129	0.040	0.251	0.311	63.89	65.078	0.719
March	6.548	6.582	0.023	(0.325)	(0.292)	100.29	101.63	0.806
April	26.575	26.671	0.056	0.307	0.368	168.96	170.23	0.801
May	32.451	32.538	0.041	0.427	0.472	178.22	179.45	0.693
June	28.996	29.131	0.082	0.430	0.509	149.08	150.13	0.625
July	29.985	30.203	0.111	0.442	0.537	118.97	119.82	0.480
August	26.019	26.246	0.135	0.429	0.554	87.258	87.678	0.266
September	-	-	-	-	-	60.000	60.000	0.000

Std. denotes the standard deviation. The initial storage is a set value. Surpluses are shown in bold in parentheses.

The minimum storage was 40 million m^3 in November, and the maximum storage occurred in May. Comparing these results with Chenari *et al.* (2014), the maximum water storage was increased by about 14.83 million m^3 and the minimum storage decreased by 9.99 million m^3 . The average coefficient of variation (i.e. the ratio of the standard deviation to the

mean) of the storage for the year (excluding the set or constrained values in September and November) was 0.006, which suggests a very high degree of consistency in the results achieved. The sustainability constraint succeeded to guide the algorithm to find solutions that store enough water over the entire year to maintain the required initial storage for the next year. This is observed clearly in the first and 13th month (September).

5.3 Fitness Function Values

The best fitness function value was $23.01 \times 10^6 \text{ m}^3$. At the solution, the value of the constraint violation penalty C was zero. The fitness function value of $23.01 \times 10^6 \text{ m}^3$ is a significant improvement (87.6%) relative to the previous value of $185.3 \times 10^6 \text{ m}^3$ in Chenari *et al.* (2014). In other words, based on these results, the total annual imbalance between the releases and demands has been reduced by 64.7%, from $f^{1/2} = 13.61 \times 10^6 \text{ m}^3$ to $f^{1/2} = 4.80 \times 10^6 \text{ m}^3$. The average number of restarts to escape stagnation of the algorithm and/or improve the results of the search (as explained in Section 2) per optimization run was 320.

Each run of the optimization algorithm took a few seconds on a personal computer (Linux, Dell OptiPlex 780, Core Duo 2, E8400 @ $2 \times 3.0 \text{ GHz}$, 8.0 GB RAM). Figure 3a illustrates the convergence of the fitness function. It can be seen that the fitness function converged within 25,000 function evaluations approximately. The algorithm's convergence is fast, which is beneficial for rapid updating of the policy of reservoir operation. The convergence point in Chenari *et al.* (2014) was 525,000 function evaluations.

As stated previously, the values of the penalty factor, $A = 100$, and exponent, $e = 2$, were determined empirically. The best alternative fitness function value was achieved with a penalty factor A of 1,000 i.e. $23.01 \times 10^6 \text{ m}^3$, based on 10 optimization runs, with $e = 2$. A safe value of 100 was therefore selected for the penalty factor A , to strike a balance that reduces the risk of premature convergence due to selection pressure; ultimately, $A = 100$ and $A = 1,000$ gave essentially the same solution, with an exponent value of $e = 2$.

Other combinations of the penalty factor, A , and exponent, including $e = 1$ and $e = 4$, gave slightly larger values of the fitness function. However, due to the effectiveness of methodology employed, consistently good results were achieved. The mean value of the fitness function, based on 10 optimization runs, ranged from $23.2 \times 10^6 \text{ m}^3$ to $23.7 \times 10^6 \text{ m}^3$ while the minimum ranged from $23.01 \times 10^6 \text{ m}^3$ to $23.04 \times 10^6 \text{ m}^3$.

Table 4 and Figure 3b provide a summary of the results of the sensitivity analysis. These results demonstrate that the formulation used is effective, stable, robust, and not overly sensitive to the values of the parameters A and e of the penalty function. It can be seen also that the parameter-free version of the penalty function, with both A and e set to unity, i.e. $A = e = 1$, is also satisfactory, albeit with a slightly lower consistency, based on the standard deviation of $0.905 \times 10^6 \text{ m}^3$. Indeed, it is interesting to note that the median and minimum values of the fitness function were effectively virtually identical.

Table 4 Influence of the parameters of the penalty function on the fitness function

A	10^0	10^1	10^2	10^3	10^4	10^2	10^0
e	2	2	2	2	2	4	1
Fitness function ($\text{m}^3 \times 10^6$)							
Min.	23.020	23.014	23.013	23.013	23.023	23.012	23.016
Median	23.037	23.038	23.024	23.039	23.041	23.042	23.037
Mean	23.383	23.725	23.381	23.186	23.558	23.451	23.454
Max.	25.333	27.368	24.319	23.978	27.884	26.943	26.281
Std.	0.682	1.368	0.475	0.309	1.522	1.166	0.905

The constraint violation, C , was zero at the solution; Std. denotes the standard deviation.

Overall, the parameter combination $(A, e) = (100, 2)$ and $(1000, 2)$ gave the best results in terms of accuracy and consistency, as can be seen in Figure 3b, with the smallest standard deviations of $0.475 \times 10^6 \text{ m}^3$ and $0.309 \times 10^6 \text{ m}^3$, respectively, in Table 4. These results (Figure 3b and Table 4) suggest that A is efficient between 100 and 1000. The present fitness function values may be compared to $185.3 \times 10^6 \text{ m}^3$ in Chenari *et al.* (2014). The results achieved here are thus a significant improvement.

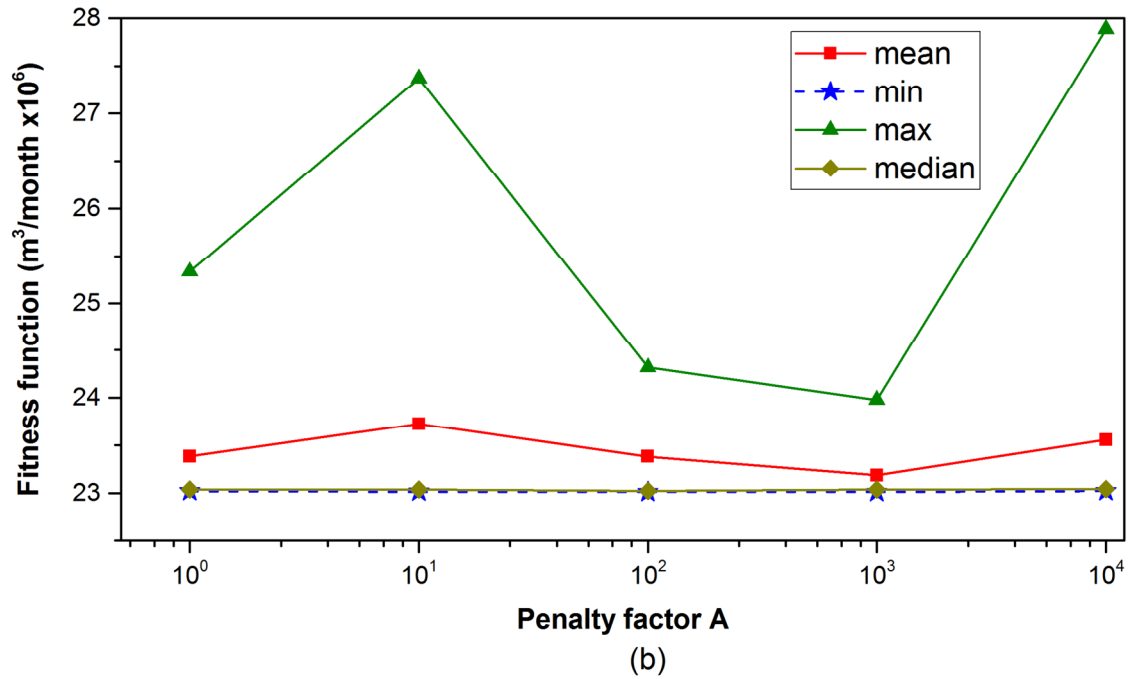
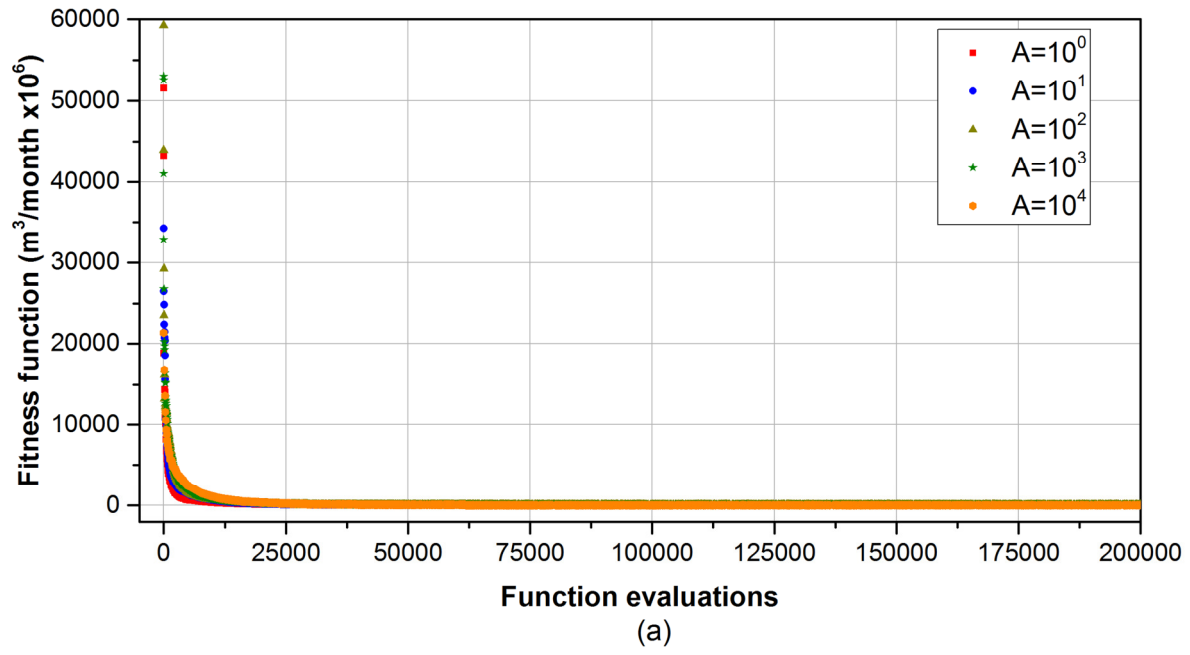


Figure 3. Properties of the fitness function. (a) Convergence characteristics (b) Accuracy and consistency. The lines in (b) are to aide visualisation. The value of the exponent in Eq. 8 is 2.

5.4 Observations on the Optimization Algorithm

Figure 4 illustrates the typical development of the decision variables of releases and storages toward the best solution during the optimization. Starting with an initial random population, it can be seen that rapid convergence was achieved within 25,000 function evaluations approximately, and the values remained stable thereafter.

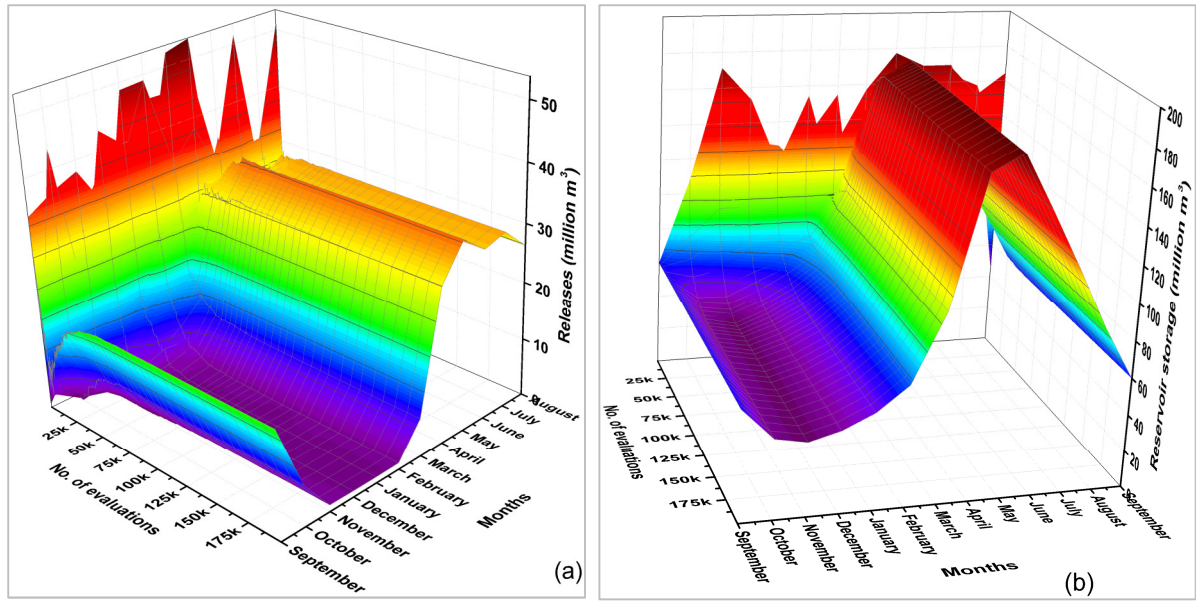


Figure 4. Evolution of the decision variables. (a) Water release (b) Reservoir storage. The irregular patterns at the far ends, at zero function evaluations, depict the initial random seeds.

Figure 5a shows the calculated percentages of the solutions in the archive, based on the respective selection probabilities of the recombination operators. PCX and UNDX were, apparently, the most successful operators, with averages of 28% and 29%, respectively. SPX and SBX had averages of 21% and 18%, respectively. The DE operator had an average of 8%. The UM operator rarely succeeded to generate dominant solutions for the archive, with an average of 0.16%.

On the other hand, Figure 5b shows the actual contributions of the various operators. All the percentages ranged between 14% and 19%. UM was the least successful operator while the most successful were PCX and UNDX followed by SPX. It can be seen that the contributions of the six recombination operators were roughly comparable.

To investigate further the relative merits of the recombination operators, Figure 6 shows heat maps of their selection probabilities for the entire optimization run. PCX generated dominant offspring in all the runs, with more solutions generated after 80,000 function evaluations. UNDX performed well in the early stages, especially before 80,000 evaluations. Then its ability to generate dominant solutions decreased slowly until the end.

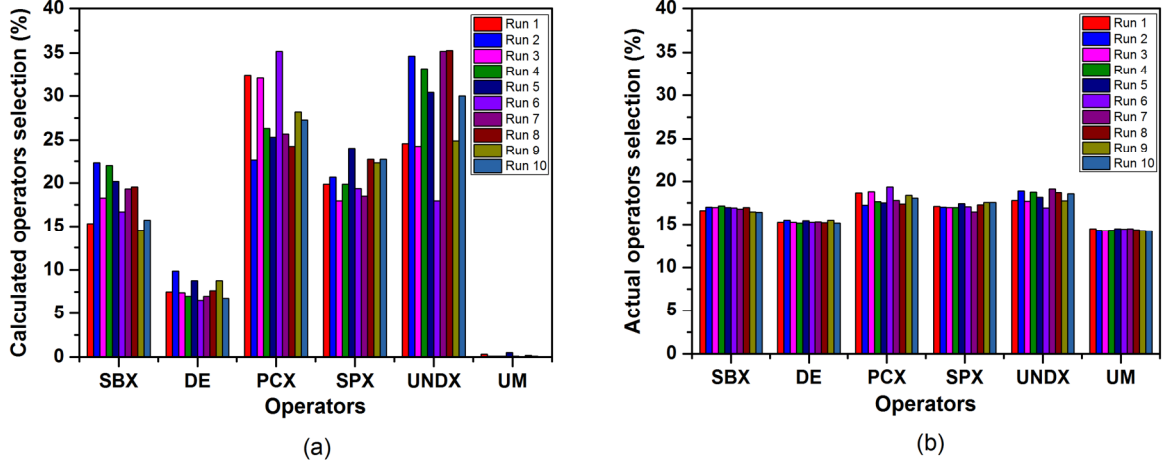


Figure 5. Relative contributions of the recombination operators (a) Calculated operator selection probabilities (b) Actual operator selection frequencies achieved

The difference between the calculated and actual operator selection probabilities may be due to the operator selection mechanism employed in Borg MOEA (Hadka and Reed 2013), i.e.

$$Q_i = \frac{C_i + \alpha}{\sum_{i=1}^K (C_i + \alpha)} ; \quad i = 1, \dots, K \quad (13)$$

where K is the number of operators; Q_i is the probability of selecting operator i ; C_i is the number of solutions produced by the i th operator in the archive; and $\alpha = 1$ is a constant used to avoid probability values of zero.

The algorithm initially sets a uniform probability of $1/K$ for all the operators. Then, the probability is updated periodically throughout the optimization. In the case of a single-objective optimization problem, the probability Q_i may remain in a limited range with no operator dominating the others because there is only one dominant solution in the archive. Therefore, the algorithm almost randomly selects the operators. On the other hand, for multi-objective problems, the algorithm generates a population of solutions in the dominance archive, and the value of Q_i changes according to Equation 13.

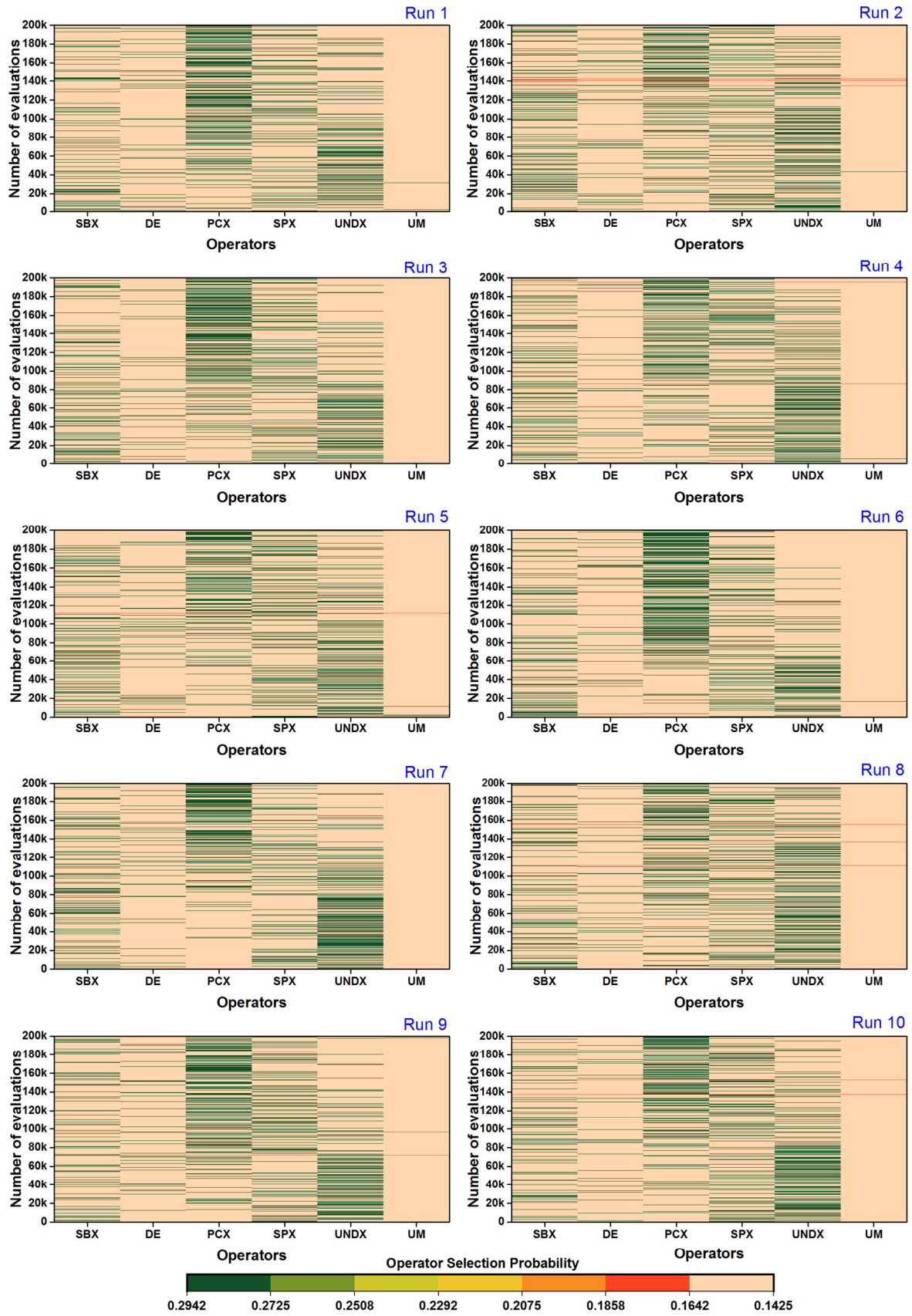


Figure 6. Selection probabilities for the recombination operators

5.5 Influence of the Reservoir Storage Sustainability Constraint

To understand the effects of the sustainability constraint, the optimization problem was also solved without the constraint on the annual reservoir drawdown, as specified originally in Chenari *et al.* (2014). The objective function value obtained was $19.97 \times 10^6 \text{ m}^3$, with mean, median and standard deviation of $20.60 \times 10^6 \text{ m}^3$, $20.63 \times 10^6 \text{ m}^3$ and $0.53 \times 10^6 \text{ m}^3$, based on 10 optimization runs. The convergence, for the best run, was achieved at 30,000 function evaluations approximately, with 399 restarts to escape from local minima and/or improve the results, subject to the total number of function evaluations allocated. The improvement achieved by reducing the value of the objective function was thus 89.2%, while the improvement achieved with the sustainability constraint in force was 87.6% (Subsection 5.3). The results achieved are summarised in Table 5.

Table 5 Reservoir operation results without the storage sustainability constraint

Month	Reservoir Release (10^6 m^3)			Storage (10^6 m^3)		
	Min.	Mean	Std.	Min.	Mean	Std.
September	18.437	18.443	0.007	60.000	60.000	0.000
October	6.863	6.876	0.009	43.184	43.203	0.010
November	1.140	1.463	0.266	40.000	40.000	0.000
December	1.035	1.374	0.255	43.070	43.965	0.529
January	1.031	1.351	0.240	49.503	51.278	1.035
February	1.104	1.401	0.211	61.190	63.775	1.507
March	6.631	6.884	0.196	96.490	99.779	1.927
April	26.797	27.014	0.160	163.94	167.83	2.306
May	32.813	33.000	0.129	171.91	176.30	2.617
June	29.465	29.628	0.096	141.22	146.07	2.875
July	30.633	30.733	0.058	109.65	114.78	3.063
August	26.754	26.798	0.022	76.282	81.575	3.174
September	-	-	-	47.458	52.797	3.215

Std. denotes the standard deviation.

The total annual imbalance between the releases and demands was reduced by 67.2%, from $f^{1/2} = 13.61 \times 10^6 \text{ m}^3$ to $f^{1/2} = 4.47 \times 10^6 \text{ m}^3$, compared to 64.7% with the sustainability constraint. Table 5 shows the reservoir operation results, without the annual reservoir storage

sustainability constraint. The maximum standard deviation of the releases was $0.266 \times 10^6 \text{ m}^3$ in November, while the smallest was $0.007 \times 10^6 \text{ m}^3$. The average coefficient of variation of the storage for the year (excluding September and November with set or constrained values) was 0.023 that demonstrates a high level of consistency in the results achieved.

6. CONCLUSIONS

A state-of-the-art evolutionary optimization algorithm (Borg MOEA) was investigated and used to solve a reservoir operation problem. The objectives of the optimization were to manage the reservoir drawdown and water releases to satisfy the requirements downstream.

The algorithm converged rapidly and reliably. For the reservoir system considered, convergence was achieved within approximately 25,000 function evaluations compared to 525,000 for a previous genetic algorithm in the literature (Chenari *et al.* 2014). The quantity of water stored in the reservoir was improved by increasing the maximum storage by 14.83 million m^3 . This has the potential to increase the economic and environmental benefits of the reservoir. The total annual imbalance between the monthly reservoir releases and water demands was reduced by 64.7%, from $13.61 \times 10^6 \text{ m}^3$ to $4.80 \times 10^6 \text{ m}^3$.

Moreover, the reservoir drawdown constraint was satisfied strictly. In other words, the required amount of water was retained in the reservoir for the next year. On the other hand, when the storage sustainability constraint was removed to conform to the original specifications of the problem considered in Chenari *et al.* (2014), the annual imbalance between the demands and releases was reduced further from $13.61 \times 10^6 \text{ m}^3$ to $4.47 \times 10^6 \text{ m}^3$ (i.e. 67.2%).

Borg MOEA deploys multiple recombination operators self-adaptively, which contributes to its effectiveness, versatility and robustness. The algorithm's performance was reliable and stable, and good results were achieved consistently and quickly, which shows, also, that the optimization model used was effective. However, for the problem considered in

this study, the algorithm seemingly did not adapt the selection of the recombination operators based on the solutions achieved by each operator. It seems the algorithm randomly selected operators to generate solutions. Additional investigation on this is thus indicated.

This research is in progress and the results achieved provide encouragement to solve even more complex real-world reservoir management problems in the future. This research could assist water managers and decision makers, and help to maximize the potential environmental and economic benefits of reservoir systems (Horne *et al.* 2016). It is vitally important to maximize the socio-economic and environmental benefits of long-term capital-intensive infrastructure such as reservoirs, at all stages including planning, design, operation, management, rehabilitation and/or upgrading. Optimization based studies can help to achieve this objective. The study provides an indicative example of the improvements that could be gained potentially by optimizing complex systems, and helps to enhance the knowledge and understanding of the dynamic properties of the system under consideration.

ACKNOWLEDGEMENT

This research was funded in part by the Government of Iraq, Ministry of Higher Education and Scientific Research (MHESR)/University of Baghdad, under the scheme of Iraqi National PhD Scholarship Programme for the first author and this is gratefully acknowledged. Also, the authors thank Professor Patrick Reed and David Hadka for providing the source code for Borg MOEA.

REFERENCES

- Barlow, E., Tanyimboh, T.T., 2014. Multiobjective memetic algorithm applied to the optimisation of water distribution systems. *Water Resour. Manag.* 28, 2229–2242. doi:10.1007/s11269-014-0608-0.
- Cancelliere, A., Giuliano, G., Nicolosi, V., Rossi, G., 2003. Optimal short-term operation of a multipurpose reservoir system under limited water supply. *Water Resour. Syst. Risk Manag. Dev.* 200–207.
- Chang, L.C., Chang, F.J., Wang, K.W., Dai, S.Y., 2010. Constrained genetic algorithms for optimizing multi-use reservoir operation. *J. Hydrol.* 390, 66–74. doi:10.1016/j.jhydrol.2010.06.031.
- Chenari, K., Ghaderi, M., Reza, M., 2014. Improved adaptive genetic algorithm and its application in short-term optimal operation of water resources system. *Agric. Environ. Management* 3, 209–217.
- Coello, C.A.C., Lamont, G.L., van Veldhuizen, D.A., 2007. *Evolutionary Algorithms for Solving Multi-Objective Problems*, 2nd ed, Genetic and Evolutionary Computation. Springer, Berlin, Heidelberg. doi:10.1007/978-0-387-36797-2.
- Coello Coello, C.A., 2002. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. *Comput. Methods Appl. Mech. Eng.* 191, 1245–1287. doi:http://dx.doi.org/10.1016/S0045-7825(01)00323-1.
- Deb, K., 2001. *Multi-Objective Optimization using Evolutionary Algorithm*, 1st edition, John Wiley & Sons Ltd.
- Deb, K., Agrawal, R.B., 1994. Simulated binary crossover for continuous search space. *Complex Systems*. doi:10.1.1.26.8485Cached.
- Deb, K., Datta, R., 2013. A bi-objective constrained optimization algorithm using a hybrid evolutionary and penalty function approach. *Eng. Optim.* 45, 503–527.

doi:10.1080/0305215X.2012.685074.

- Deb, K., Joshi, D., Anand, A., 2002. Real-coded evolutionary algorithms with parent-centric recombination. *Proc. 2002 Congr. Evol. Comput. CEC 2002* 1, 61–66. doi:10.1109/CEC.2002.1006210.
- Deb, K., Manikanth, M., Shikhar, M., 2003. A fast multi-objective evolutionary algorithm for finding well-spread pareto-optimal solutions. KanGAL Technical Report No.2003002, IIT, Kanpur, India.
- Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* 6, 182–197. doi:10.1109/4235.996017.
- Deb, K., Tiwari, S., 2005. Multi-objective optimization of a leg mechanism using genetic algorithms. *Eng. Optim.* 37, 325–350. doi:10.1080/03052150500066695.
- Dridi, L., Parizeau, M., Mailhot, A., Villeneuve, J.P., 2008. Using evolutionary optimization techniques for scheduling water pipe renewal considering a short planning horizon. *Computer-Aided Civil and Infrastructure Engineering* 23, 625–635. doi:10.1111/j.1467-8667.2008.00564.x
- Eskandar, H., Sadollah, A., Bahreininejad, A., Hamdi, M., 2012. Water cycle algorithm – A novel metaheuristic optimization method for solving constrained engineering optimization problems. *Comput. Struct.* 110–111, 151–166. doi:http://dx.doi.org/10.1016/j.compstruc.2012.07.010.
- Feng, B.C., Liu, L., Member, A., 1999. Using genetic algorithms to solve construction time-cost trade-off problems. *J. Comput. Civ. Eng.* 10, 184–189.
- Formiga, K.T.M., Chaudhry, F.H., Cheung, P.B., Reis, L.F.R., 2003. Optimal design of water distribution system by multiobjective evolutionary methods, in: Fonseca, C.M., Fleming, P.J., Zitzler, E., Thiele, L., Deb, K. (Eds.), *Evolutionary Multi-Criterion Optimization*:

- Second International Conference, EMO 2003, Faro, Portugal, April 8-11, 2003. Proceedings. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 677–691. doi:10.1007/3-540-36970-8_48.
- Hadka, D., Reed, P., 2013. Borg: An auto-adaptive many-objective evolutionary computing framework. *Evol. Comput.* 21, 1–30. doi:10.1162/EVCO_a_00075.
- Hadka, D., Reed, P., 2012. Diagnostic assessment of search controls and failure modes in many-objective evolutionary optimization. *Evol. Comput.* 20, 423–452. doi:10.1162/EVCO_a_00053.
- Hadka, D., Reed, P.M., Simpson, T.W., 2012. Diagnostic assessment of the borg MOEA for many-objective product family design problems, in: 2012 IEEE Congress on Evolutionary Computation, CEC 2012. pp. 10–15. doi:10.1109/CEC.2012.6256466.
- Haimes, Y.Y., Hall, W.A., 1977. Sensitivity, responsivity, stability and irreversibility as multiple objectives in civil systems. *Adv. Water Resour.* 1, 71–81. doi:10.1016/0309-1708(77)90025-2.
- Horne, A., Szemis, J.M., Kaur, S., Webb, J.A., Stewardson, M.J., Costa, A., Boland, N., 2016. Optimization tools for environmental water decisions: A review of strengths, weaknesses, and opportunities to improve adoption. *Environ. Model. Softw.* 84, 326–338. doi:http://dx.doi.org/10.1016/j.envsoft.2016.06.028.
- Jothiprakash, V., Shanthi, G., 2006. Single reservoir operating policies using genetic algorithm. *Water Resour. Manag.* 20, 917–929. doi:10.1007/s11269-005-9014-y.
- Kim, T., Heo, J.-H., 2006. Multireservoir system optimization in the Han River Basin using multi-objective genetic algorithms. *Hydrol. Process.* 20, 2057–2075. doi:10.1007/BF02830091.
- Kita, H., Ono, I., Kobayashi, S., 2000. Multi-parental extension of the unimodal normal distribution crossover for real-coded genetic algorithms. *Trans. Soc. Instrum. Control*

- Eng. 36, 875–883. doi:10.9746/sicetr1965.36.875.
- Knowles, J., Corne, D., 2002. On metrics for comparing nondominated sets, in: *Evolutionary Computation*, 2002. CEC '02. Proceedings of the 2002 Congress on. pp. 711–716. doi:10.1109/CEC.2002.1007013.
- Kollat, J.B., Reed, P.M., 2006. Comparing state-of-the-art evolutionary multi-objective algorithms for long-term groundwater monitoring design. *Adv. Water Resour.* 29, 792–807. doi:10.1016/j.advwatres.2005.07.010.
- Lobo, F.G., Lima, C.F., Michalewicz, Z., 2007. *Parameter Setting in Evolutionary Algorithms*, Springer.
- Michalewicz, Z., Logan, T., Swaminathan, S., 1994. Evolutionary operators for continuous convex parameter spaces. *Proc. 3rd Annu. Conf. Evol. Program.* 84–97.
- Reddy, M.J., Kumar, D.N., 2006. Optimal reservoir operation using multi-objective evolutionary algorithm. *Water Resour. Manag.* 20, 861–878. doi:10.1007/s11269-005-9011-1.
- Reed, P.M., Hadka, D., Herman, J.D., Kasprzyk, J.R., Kollat, J.B., 2013. Evolutionary multiobjective optimization in water resources: The past, present, and future. *Adv. Water Resour.* 51, 438–456. doi:10.1016/j.advwatres.2012.01.005.
- Régnier, J., Sareni, B., Roboam, X., 2005. System optimization by multiobjective genetic algorithms and analysis of the coupling between variables, constraints and objectives. *COMPEL - Int. J. Comput. Math. Electr. Electron. Eng.* 24, 805–820. doi:10.1108/03321640510598157.
- Saleh, S., Tanyimboh, T.T., 2016. Multi-directional maximum-entropy approach to the evolutionary design optimization of water distribution systems. *Water Resour. Manag.* 30, 1885–1901. doi:10.1007/s11269-016-1253-6.
- Saleh, S.H.A., Tanyimboh, T.T., 2014. Optimal design of water distribution systems based on

- entropy and topology. *Water Resour. Manag.* 28, 3555–3575. doi:10.1007/s11269-014-0687-y.
- Saleh, S.H.A., Tanyimboh, T.T., 2013. Coupled topology and pipe size optimization of water distribution systems. *Water Resour. Manag.* 27, 4795–4814. doi:10.1007/s11269-013-0439-4.
- Scola, L.A., Takahashi, R.H.C., Cerqueira, S.A.A.G., 2014. Multipurpose water reservoir management: an evolutionary multiobjective optimization approach. *Mathematical Problems in Engineering* 2014 doi:10.1155/2014/638259.
- Sharif, M., Wardlaw, R., 2000. Multireservoir systems optimization using genetic algorithms: case study. *J. Comput. Civ. Eng.* 14, 255–263.
- Sierra, M.R., Coello Coello, C.A., 2005. Improving PSO-based multi-objective optimization using crowding, mutation and ϵ -dominance, in: Coello Coello, C.A., Hernández Aguirre, A., Zitzler, E. (Eds.), *Evolutionary Multi-Criterion Optimization: Third International Conference, EMO 2005, Guanajuato, Mexico, March 9-11, 2005. Proceedings.* Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 505–519. doi:10.1007/978-3-540-31880-4_35.
- Siew, C., Tanyimboh, T.T., 2012. Penalty-free feasibility boundary convergent multi-objective evolutionary algorithm for the optimization of water distribution systems. *Water Resour. Manag.* 26, 4485–4507. doi:10.1007/s11269-012-0158-2.
- Siew, C., Tanyimboh, T.T., Seyoum, A.G., 2014. Assessment of penalty-free multi-objective evolutionary optimization approach for the design and rehabilitation of water distribution systems. *Water Resour. Manag.* 28, 373–389. doi:10.1007/s11269-013-0488-8.
- Siew, C., Tanyimboh, T.T., Seyoum, A.G., 2016. Penalty-free multi-objective evolutionary approach to optimization of Anytown water distribution network. *Water Resour. Manag.*

- 30, 3671–3688. doi:10.1007/s11269-016-1371-1.
- Storn, R., Price, K., 1997. Differential evolution -- a simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* 11, 341–359. doi:10.1023/A:1008202821328.
- Tanyimboh, T.T., Seyoum, A.G., 2016. Multiobjective evolutionary optimization of water distribution systems: Exploiting diversity with infeasible solutions. *J. Environ. Manage.* 183, Part, 133–141. doi:http://dx.doi.org/10.1016/j.jenvman.2016.08.048.
- Tilmant, A., Fortemps, P., Vanclooster, M., 2002. Effect of averaging operators in fuzzy optimization of reservoir operation. *Water Resour. Manag.* 16, 1–22. doi:10.1023/A:1015523901205.
- Tract, A.B., 1997. Multiobjective optimization design with pareto genetic algorithm. *Struct. Eng.* 123, 1252–1261.
- Tsutsui, S., Yamamura, M., Higuchi, T., 1999. Multi-parent recombination with simplex crossover in real coded genetic algorithms, in: *Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation-Volume 1*. pp. 657–664.
- Vrugt, J., Robinson, B., 2007. Improved evolutionary optimization from genetically adaptive multimethod search. *Proc. Natl. Acad. Sci.* 104, 708–711. doi:10.1073/pnas.0610471104.
- Woldesenbet, Y.G., Yen, G.G., Tessema, B.G., 2009. Constraint handling in multiobjective evolutionary optimization. *IEEE Trans. Evol. Comput.* 13, 514–525. doi:10.1109/TEVC.2008.2009032.
- Wu, Y.G., Zou, J., 2012. An advanced multi-objective genetic algorithm based on borda number, in: *Progress in Industrial and Civil Engineering, Applied Mechanics and Materials*. Trans Tech Publications, pp. 4909–4915. doi:10.4028/www.scientific.net/AMM.204-208.4909.

- Yang, J., Soh, C.K., 1997. Structural optimization by genetic algorithms with tournament selection. *J. Comput. Civ. Eng.* 11, 195–200. doi:10.1061/(ASCE)0887-3801(1997)11:3(195).
- Zhang, Q., Liu, W., Li, H., 2009. The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances, in: 2009 IEEE Congress on Evolutionary Computation. pp. 203–208. doi:10.1109/CEC.2009.4982949.
- Zheng, F., Simpson, A.R., Zecchin, A.C., Maier, H.R., Feifei, Zh., 2016. Comparison of the searching behavior of NSGA-II, SAMODE, and Borg MOEAs applied to water distribution system design problems. *J. Water Resour. Plan. Manag.* 142. doi:10.1061/(ASCE)WR.1943-5452.0000650.
- Zitzler, E., Laumanns, M., Thiele, L., 2002. SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization, in: *Evolutionary Methods for Design, Optimisation, and Control*. CIMNE, Barcelona, Spain, pp. 95–100.

Figure Captions

Figure 1. Graphical representation of the ϵ -progress concept in a minimization problem with two objectives. Solutions (1) and (2) are new solutions in unoccupied boxes and thus represent improvements. Solution (3) is not considered as an improvement because it resides in a previously occupied box. The shaded boxes were previously occupied while the unshaded boxes were not previously occupied (Hadka and Reed 2013).

Figure 2. Reservoir releases and storage (a) Chenari *et al.* (2014) (b) Present formulation with sustainability constraint (c) Present formulation without sustainability constraint

Figure 3. Properties of the fitness function. (a) Convergence characteristics (b) Accuracy and consistency. The lines in (b) are to aide visualisation. The value of the exponent in Eq. 8 is 2.

Figure 4. Evolution of the decision variables. (a) Water release (b) Reservoir storage. The irregular patterns at the far ends, at zero function evaluations, depict the initial random seeds.

Figure 5. Relative contributions of the recombination operators (a) Calculated operator selection probabilities (b) Actual operator selection frequencies achieved

Figure 6. Selection probabilities for the recombination operators

Table Captions

Table 1 Default values of the parameters used in Borg MOEA

Table 2 Reservoir inflows and water demands (Chenari *et al.* 2014)

Table 3 Reservoir operation results based on ten optimization runs

Table 4 Influence of the parameters of the penalty function on the fitness function

Table 5 Reservoir operation results without the storage sustainability constraint